

# The Embedded Muse 169

Copyright 2008 TGG

November 17, 2008

You may redistribute this newsletter for noncommercial purposes. For commercial use contact [info@ganssle.com](mailto:info@ganssle.com). Subscribe and unsubscribe info is at the end of this email.

EDITOR: Jack Ganssle ([jack@ganssle.com](mailto:jack@ganssle.com))

## CONTENTS:

- Editor's Notes
- Microchip's 8 Bit Business
- Book Review
- Funny Datasheets
- Header Guards
- Jobs!
- Joke for the Week
- About The Embedded Muse

\*\*\*\*\*

This issue of The Embedded Muse is sponsored by Netrino.

Coding standards and code reviews are important tools for embedded software teams. Netrino Institute offers training courses on both subjects. This winter, Netrino will give the first public session of "Developing Effective Coding Standards". For course locations and dates for this as well as the next "Embedded Software Boot Camp" see <http://www.NetrinoInstitute.com/Calendar-Muse> .

\*\*\*\*\*

## Editor's Notes

Did you know it IS possible to create accurate schedules? Or that most projects consume 50% of the development time in debug and test, and that it's not hard to slash that number drastically? Or that we know how to manage the quantitative relationship between complexity and bugs? Learn this and far more at my Better Firmware Faster class, presented at YOUR facility. See <http://www.ganssle.com/classes.htm> .

*Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**

Computerworld's annual salary survey is out:

<http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=326024> . It's for IT workers rather than embedded engineers, but the data is interesting nonetheless.

```
switch(city){
case San Jose:
case Phoenix:
case Austin:
```

```
Are you in the San Jose, Phoenix or Austin areas? I'll present a public
version of the Better Firmware Faster class in San Jose on December 8,
Phoenix on December 10 and Austin on December 12. Registration and
other info here: http://www.ganssle.com/classes.htm . You'll earn 0.7
Continuing Education Units, learn a lot, and have more than a little
fun. Sign up before November 8 and receive a $50.00 discount.
}
```

## **Microchip's 8 Bit Business**

As many readers know, Microchip and ON Semiconductor have made an offer to purchase Atmel. That offer was rejected by Atmel's board, though Microchip says they intend to continue to pursue the acquisition.

I wrote about this here: <http://embedded.com/columns/breakpoint/211600577> . That piece birthed a storm of calls from analysts who follow the semiconductor industry. One of them – a name everyone would recognize – sent me a new report that downgrades their previous “neutral” rating on Microchip to a “sell.” Among the reasons listed is an anticipated decline in sales of 8 bit microprocessors as prices of 32 bitters fall and erode the low end of the market.

That reasoning disturbs me.

There's no doubt that 32 bit prices will continue to fall. Consider Luminary Micro's ARM controllers. They, like ST, Atmel and others, are doing a remarkable job of positioning the ARM as the new century's 8051. Costs keep coming down; some of these parts are available for under a buck.

But does that mean 8 bit sales will falter?

Excluding the hopefully short-term fallout from the financial mess currently rocking the world, I think not. “Transistors are free” is our mantra, but of course that's not really true. Transistors do have a cost, both in dollars (well, microcents), size, and power

*Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

consumption. With similar technology 8 bit controllers will always outperform a larger CPU in these categories.

But as 32 bit devices become cheaper, so do the 8 biters. The most important lesson we embedded people have learned since the first microprocessor appeared is that dropping costs create new and often unimagined applications. Prior to the 4004 computers were useful in only a relative handful of high-end science and business arenas. Now the iPhone – which even Chester Gould didn't anticipate – exists only due to cheap computing. That's true for millions of products.

8 bit microcontrollers will transcend packaging, somehow. We will quite literally sow them into the environment, 21<sup>st</sup> century Johnny Appleseeds scattering sensor networks by the millions. These insanely low cost and low power computers will harvest power from the environment. Who knows? Maybe farmers will sow them with their seeds to monitor or even enhance the growth of each stalk of corn. Or maybe each kernel.

And don't forget capital costs. For better or worse a lot of the benefits of 32 bit parts come from process shrinks. But each shrink comes at astronomical costs. Many 8 bit parts still come off 250 nm production lines that have long been expensed. A 65 nm fab that costs \$5B has to be depreciated. Fabless or not, vendors relying on state-of-the art processes incur a cost associated with that depreciation.

I've been hearing for 15 years that 8 bits is "dead." It's not. \$6B worth of eight bit processors are sold each year. I predict a lot of growth, and tremendously exciting new applications.

## **Book Review**

The kind folks at Newnes sent me a copy of Joseph Yiu's 2007 book "The Definitive Guide to the ARM Cortex-M3," which is, as the title suggests, all about ARM's newest embedded processor.

The Cortex-M3 is an important departure for ARM. It's tightly focused on embedded applications, particularly for microcontrollers. As such there's no cache (though licensees, the vendors who translate ARM IP into actual microcontrollers, can add it). Cache, while a nice speed enhancer, always brings significant determinism problems to real-time systems. The Cortex-M3 does not support the ARM instruction set, using Thumb-2 instructions to enhance code density. This architecture also includes a very sophisticated interrupt controller well-tailored to the needs of deeply-embedded real-time systems.

*Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

Given the importance of the Cortex-M3 this book fills an essential void. Sure, there's more detail available in data from ARM, but those data sheets are so huge it's hard to get a general sense of the nature of the processor.

Alas, the book has numerous typos and grammatical mistakes. Possibly the most egregious is in a sidebar which attempts to explain the difference between "bit-banding" and "bit-banging," where the author mixes up the two terms and defines bit-banding as the software control of a hardware pin to simulate a UART. But in general the errors are more annoying than important.

I found the stream of "this might be" references to features a bit frustrating, but in fact the author is correct in his hesitancy. For ARM sells IP, not hardware, and it's up to the licensees to decide which features go into a particular device. And there's little on I/O for the same reason.

Unlike ARM's datasheets, the book includes very useful chapters on porting code from the ARM7 and setting up a development environment using Keil and GNU toolchains. For many developers the latter alone will make the book worthwhile.

An appendix lists the instruction set. Some instructions have very concise and detailed descriptions. Others, unfortunately, don't.

If you're not planning to use the Cortex-M3, but want to get insight into what will be an important arena in the embedded world, this book is the best resource I know. If you are using that architecture, it's sort of a "Cortex-Lite" but very useful introduction that you'll have to augment with the datasheets.

## **Funny Datasheets**

In keeping with the occasional series of funny datasheets, Bob Paddock sent this link: <http://datasheets.maxim-ic.com/en/ds/MAXQ1103.pdf> . It's the spec for a "High-Performance Secure RISC Microcontroller." It's pretty secure, all right. The two-page summary gives, at the end, a link to the complete datasheet... which takes one right back to the beginning of the page!

## **Header Guards**

Steve Strobel wrote about header guards: I have adopted an unconventional include guard to detect and warn about such header include loops. It is even bigger and uglier than the standard include guard (which I think is ugly), but has saved me lots of frustration by

*Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

letting me know when a cycle exists before I waste time trying to track down a bunch of related compiler errors.

```
//-----  
#ifndef main_h_include_finished  
#elif defined(main_h_include_started)  
    #warning "HEADER FILE INCLUDE LOOP!!!"  
#else  
    #define main_h_include_started  
//-----  
  
    // header file contents go here  
  
//-----  
    #define main_h_include_finished  
#endif  
//-----
```

## **Jobs!**

Let me know if you're hiring firmware or embedded designers. No recruiters please, and I reserve the right to edit ads to fit the format and intents of this newsletter. Please keep it to 100 words.

Despite the downturn, embedded systems consultancy Netrino (<http://www.netrino.com>) is still growing fast. The company currently has open positions for embedded software developers as well as digital and analog electronics designers in many parts of the U.S. The company's mix of development, consulting, and training work will entertain your brain and makes for a collegial learning environment. If you may be interested, please send your current resume to [careers@netrino.com](mailto:careers@netrino.com).

Oztek in New Hampshire is looking for a self-starter to develop embedded control system for motor drives and other power related applications. Responsibilities:

- Elicit and mature control requirements
- Software estimation and planning
- Lead quick-turn feasibility studies for risk reduction
- Develop real-time software architectures
- Perform simulation using tools such as MATLAB
- Develop and implement DSP power system control algorithms
- Write concise documentation
- Participate in the design review process

Qualifications:

*Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

- BSEE / BSCS and 5+ years
- Fluent in C
- Ability to read schematics
- Comfortable using oscilloscopes, multimeters, and JTAG emulators
- Exposure to Agile development methodologies
- Power systems domain knowledge

Respond to [softwarejobs@oztekcorp.com](mailto:softwarejobs@oztekcorp.com).

## **Joke for the Week**

Fifty Ways to Hose Your Code [with apologies to Paul Simon]

The problem's all inside your code she said to me;  
Recursion is easy if you take it logically.  
I'm here to help you if you're struggling to learn C,  
There must be fifty ways to hose your code.

She said it's really not my habit to include,  
And I hope my files won't be lost or misconstrued;  
But I'll recompile at the risk of getting screwed,  
There must be fifty ways to hose your code.

Just blow up the stack, Jack,  
Make a bad call, Paul,  
Just hit the wrong key, Lee,  
And set your pointers free.

Just mess up the bus, Gus,  
You don't need to recurse much,  
You just listen to me.

She said it grieves me to see you compile again.  
I wish there were some hardware that wasn't such a pain.  
I said I appreciate that and could you please explain,  
About the fifty ways.

She said why don't we both just work on it tonight,  
And I'm sure in the morning it'll be working just right.  
Then she hosed me and I realized she probably was right,  
There must be fifty ways to hose your code.

*Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

Just lose the address, Les,  
Clear the wrong Int, Clint,  
Traverse the wrong tree, Lee,  
And set your list free.

Just mess up the bus, Gus,  
You don't need to recurse much,  
You just program in C.

## **About The Embedded Muse**

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at [jack@ganssle.com](mailto:jack@ganssle.com).

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.

*Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**